COMPLIANCE COMPONENT

Last Updated: 6/07/2006

DEFINITION	
Name	Extract Transform and Load (ETL) Design
Description	This document will address specific design elements that must be resolved before the ETL process can begin. These include determining: • Whether it is better to use an ETL suite of tools or hand-code the ETL process with available resources. • If batch processing will provide the data in a timely manner. • How much of the ETL process will be automated with schedulers, alert notifications and work flow procedures.
Rationale	Certain design elements are a fundamental and necessary first decision in the development of an ETL system. These choices affect everything and a change in these elements can mean implementing the entire system over again from the very start. The key to applying these design elements is to apply them consistently.
Benefits	By addressing these design elements, we ensure that the ETL system can do the following: • Deliver data most effectively to end user tools • Add value to data in the cleaning and conforming steps • Protect and document the lineage of data

ASSOCIATED ARCHITECTURE LEVELS	
Specify the Domain Name	Information
Specify the Discipline Name	Knowledge Management
Specify the Technology Area Name	Extract Transform and Load (ETL)
Specify the Product Component Name	

COMPLIANCE COMPONENT TYPE	
Document the Compliance Component Type	Guideline
Component Sub-type	

COMPLIANCE DETAIL	
State the Guideline, Standard or Legislation	**Proven Technology** Depending on the scope of the ETL process, a decision must be made as to whether an ETL software suite will be used or the process will be 'hand-coded' using available resources. If the scope of the project is large, purchasing an ETL software suite will ultimately reduce the cost of building and maintaining the ETL process. Although the ETL process can be created as a 'hand-coded' process, there are advantages to using purchased ETL tools. • Simpler, faster, less expensive development • Simpler connectivity to a wide variety of complex sources such as SAP applications and/or mainframes. • Parallel pipe-lined multithreaded operation • ETL tools can be used effectively by less skilled staff

- Many ETL tools have integrated metadata repositories that can synchronize metadata from source systems, target databases and other business intelligence tools.
- Most ETL tools automatically generate metadata at every step in the process and enforce a consistent metadata-driven methodology.
- Most ETL tools have a comprehensive built-in scheduler aiding in documentation, ease of creation, and management change.
- The metadata repository of most ETL tools can automatically produce data lineage (looking backward) and data dependency analysis (looking forward).
- ETL tools should be able to handle all forms of complex data type conversions.
- Most ETL tools deliver good performance for very large data stores. If the ETL data volume is or is expected to become large, an ETL suite of tools is recommended.
- Most ETL tools will perform an automatic change-impact analysis for downstream processes and applications that are affected by a proposed schema change.
- An ETL-tool approach can be augmented with selected processing modules hand-coded in an underlying programming language.
- A proven ETL tool suite can help you avoid reinventing the wheel. These tools are designed for what you are trying to do- provide usable data to the end user.

For smaller ETL projects, hand-coded projects may be quicker, cheaper and more flexible. Some of the advantages of hand-coding an ETL system are:

- A purchased tool-based approach will limit you to the tool vendor's abilities and their unique scripting language. However, all ETL tools allow *escapes* to standard programming languages in isolated modules.
- Hand-coded ETL provides unlimited flexibility, if that is indeed what you need.
- Automated unit testing tools are available in a hand-coded system but not with a tool-based approach.
- You can more directly manage metadata in hand-coded systems, although at the time you must create all your own metadata interfaces.
- A brief requirements analysis of an ETL system quickly points you toward file-based processing, not database-stored procedures. File-based processes are more direct. They're simply coded, easily tested and well understood.

Batch vs. Streaming Data Flow

The standard design for an ETL system is based on periodic batch extracts from the source data, which then flows through the system, resulting in a batch update to the data exported from the ETL system. However, when the real-time nature of the data exported becomes sufficiently urgent, it may be necessary to implement a streaming data flow in which the data at the record level continuously flows from the extraction process to the data exported from the system.

Scheduler Automation

It must be determined how deeply to control the overall ETL system with automated scheduler technology. At one extreme, all jobs are manually controlled and executed. At the other extreme, a master scheduler tool manages all the ETL jobs, statuses, alerts and flow processes.

	Exception Handling Exception handling should not be a random series of alerts or comments placed in files but rather should be a system-wide, uniform mechanism for reporting all instances of exceptions created by the ETL processes into a single database, with the name of the process, the time of the exception, its initially diagnosed severity, the action subsequently taken and the ultimate resolution status of the exception. **Quality Handling** All quality problems need to generate an audit record attached to the final dimension or fact data. Corrupted or suspected data needs to be handled with a small number of uniform responses, such as filling in missing text data with a question mark or supplying least biased estimators of numeric values that exist. **Recovery & Restart** You need to build your ETL system around the ability to recover from abnormal ending of a job and restart. ETL jobs need to be reentrant, otherwise impervious to incorrect multiple updating. **Metadata** In ETL, a metadata repository is where all the metadata information about source, target, transformations, mapping, workflows, sessions etc, are stored. From this repository, metadata can be manipulated, queried and retrieved with the help of wizards provided by metadata capturing tools. During the ETL process, when we are mapping source and target systems, we are actually mapping their metadata. A useful metadata fact stored in a repository can be a handy resource to know about the organization's data systems. Assume that each department in an organization may have different business definitions, data types, attribute names for the same attribute or they may have a single business definition for many attributes. These anomalies can be overcome by properly maintaining metadata for these attributes in the centralized repository. Thus, metadata plays a vital role in explaining about how, why, where data can be found, retrieved, stored and used efficiently in an information management system. **Security** Physical and administrative safeguards need to surround every on-line table and backup tape in the ETL environment. Archived data sets should be stored with checksums to verify that they have not been altered in any way.
	The Data Warehouse ETL Toolkit by Ralph Kimball, Wiley Publishing Inc, 2004

Compliance Sources

Name	Data Modeling	*Website*	www.learningdatamodeling.com
Contact Information			
Name		*Website*	
Contact Information			

KEYWORDS

List Keywords	Extract, Transform, Data Load, Extract Transform and Load (ETL), ETL tool suite, Batch processing, Scheduler automation, Exception handling, Quality handling, Recovery, Restart, Metadata, Security

COMPONENT CLASSIFICATION

Provide the Classification	☐ *Emerging*	☒ *Current*	☐ *Twilight*	☐ *Sunset*
Sunset Date				

COMPONENT SUB-CLASSIFICATION		
Sub-Classification	Date	Additional Sub-Classification Information
☒ *Technology Watch*	6-07-06	Traditional ETL approaches rely on proprietary ETL engines deployed between sources and targets. The functionality of relational databases is rapidly eliminating the ETL category by incorporating transformation functionalities. This is creating a new process ELT (extract load and transform) where all complex processing of data occurs inside the database itself. See ELT Reference document.
☐ *Variance*		
☐ *Conditional Use*		

Rationale for Component Classification	
Document the Rationale for Component Classification	

Migration Strategy	
Document the Migration Strategy	

Impact Position Statement	
Document the Position Statement on Impact	

CURRENT STATUS				
Provide the Current Status	☐ *In Development*	☐ *Under Review*	☒ *Approved*	☐ *Rejected*

AUDIT TRAIL			
Creation Date	4/21/2006	*Date Approved / Rejected*	6/13/06
Reason for Rejection			
Last Date Reviewed		*Last Date Updated*	
Reason for Update			